

Node Implementation

9 min

Let's implement a linked list in Python. As you might recall, each linked list is a sequential chain of nodes. So before we start building out the `LinkedList` itself, we want to build up a `Node` class in Python that we can use to build our data containers.

Remember that a node contains two elements:

- data
- a link to the next node

Ready? Let's get started!

Note: Because the workspace is set up with spaces instead of tabs, you will need to use spaces to prevent Python from throwing an error. You can learn more about this [here](#).

Instructions

1.

Within **script.py** in the pane to the right, create an empty `Node` class.

Inside, define an `__init__()` method for the `Node`. It should take a `value` and a `next_node`.

`next_node` should default to `None` if not provided. These variables should be saved to `self` with corresponding key names.

Hint

Remember, to create a Python class with an `__init__()` method:

```
class SomeClass:
    def __init__(self, some_attribute):
        self.some_attribute = some_attribute
```

In the case of `Node`, the passed in attributes would be `value` and `next_node`.

2.

Define `.get_value()` and `.get_next_node()` methods. These should return the corresponding values from `self`.

Hint

For example, if you wanted to access the value of `stuff` using `self`:

```
def get_stuff(self):  
    return self.stuff
```

3.

Define a `.set_next_node()` method that takes `self` and `next_node` as parameters and allows you to update the link to the next node.

Hint

A setter method would look something like:

```
def set_age(self, new_age):  
    self.age = new_age
```

4.

Outside the `Node` class, create an instance of `Node` called `my_node` with a value of 44.

Use `.get_value()` to print the value of `my_node`.

script.py

```
# Define your Node class below:  
  
class Node():  
    def __init__(self, value, next_node = None):  
        self.value = value  
        self.next_node = next_node  
  
    def get_value(self):  
        return self.value  
  
    def get_next_node(self):  
        return self.next_node  
  
    def set_next_node(self, next_node):  
        self.next_node = next_node  
  
my_node = Node(44)  
  
print(my_node.get_value)
```